

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re U.S. Patent Application)
Applicant: Yoshida et al.)
Serial No.)
Filed: March 15, 2004)
For: DEADLOCK PRE-DETECTION)
PROGRAM)
Art Unit:)

I hereby certify that this paper is being deposited with the United States Postal Service as EXPRESS MAIL in an envelope addressed to: Mail Stop PATENT APPLICATION, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on this date.

March 15, 2004 *David Owen*
Date Express Mail Label No.: EV032735771 US

CLAIM FOR PRIORITY

Mail Stop PATENT APPLICATION
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:


Applicants claim foreign priority benefits under 35 U.S.C. § 119 on the basis of the foreign application identified below:

Japanese Patent Application No. 2003-358268, filed October 17, 2003.

A certified copy of the priority document is enclosed.

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By 
Patrick G. Burns
Registration No. 29,367

Customer No. 24978

March 15, 2004
300 South Wacker Drive - Suite 2500
Chicago, Illinois 60606
Phone: (312) 360-0080
Fax: (312) 360-9315
P:\DOCS\3169\70025\472719.DOC



071734

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 1 0 月 1 7 日
Date of Application:

出 願 番 号 特 願 2 0 0 3 - 3 5 8 2 6 8
Application Number:

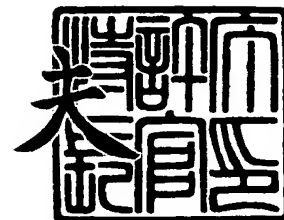
[ST. 10/C]: [J P 2 0 0 3 - 3 5 8 2 6 8]

出 願 人 富 士 通 株 式 会 社
Applicant(s):

2 0 0 3 年 1 2 月 1 7 日

特 許 庁 長 官
Commissioner,
Japan Patent Office

今 井 康



出 証 番 号 出 証 特 2 0 0 3 - 3 1 0 4 8 0 4



【書類名】 特許願
【整理番号】 0350684
【あて先】 特許庁長官殿
【国際特許分類】 G06F 9/46
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号富士通株式会社内
 【氏名】 吉田 敦
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号富士通株式会社内
 【氏名】 富永 斉
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号富士通株式会社内
 【氏名】 中村 智洋
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号富士通株式会社内
 【氏名】 則竹 泰典
【特許出願人】
 【識別番号】 000005223
 【氏名又は名称】 富士通株式会社
【代理人】
 【識別番号】 100089244
 【弁理士】
 【氏名又は名称】 遠山 勉
【選任した代理人】
 【識別番号】 100090516
 【弁理士】
 【氏名又は名称】 松倉 秀実
 【連絡先】 0 3 - 3 6 6 9 - 6 5 7 1
【手数料の表示】
 【予納台帳番号】 012092
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9705606

**【書類名】 特許請求の範囲****【請求項 1】**

コンピュータに、

複数のデータベースのうちのいずれかへのアクセスを伴うアクセスステップを含む複数の処理ステップにより構成される業務ロジック設計情報を読み込ませる第 1 手順、

前記業務ロジック設計情報に基づいて、少なくとも 2 つのアクセスステップにより構成される処理ルートを生成する第 2 手順、

前記処理ルートから、第 1 アクセスステップ、及び第 2 アクセスステップを取得する第 3 手順、

前記第 1 及び第 2 アクセスステップそれぞれによるデータベースへのアクセス順序が予め定められたアクセス順序か否かを判定する第 4 手順、

前記アクセス順序が予め定められたアクセス順序でないと判定された場合に、予め定められたアクセス順序から逸脱した旨を報知する第 5 手順、
を実行させるためのデッドロック事前検出プログラム。

【請求項 2】

前記業務ロジック設計情報は、前記アクセスステップ及び分岐条件ステップを含む複数の処理ステップにより構成され、

前記第 2 手順は、前記業務ロジック設計情報に基づいて、少なくとも 2 つのアクセスステップにより構成される少なくとも 2 つの処理ルートを生成し、

前記第 3 手順は、各処理ルートごとに、その処理ルートから、第 1 アクセスステップ、及び第 2 アクセスステップを取得し、

前記第 4 手順は、各処理ルートごとに、前記第 1 及び第 2 アクセスステップによるデータベースへのアクセス順序が予め定められたアクセス順序か否かを判定する、
請求項 1 に記載のデッドロック事前検出プログラム。

【請求項 3】

前記複数のデータベース相互間の対応関係を表す対応関係データを読み込ませる第 6 手順、

前記対応関係データに基づいて、前記予め定められたアクセス順序を生成する第 7 手順、
をさらに備える、
請求項 1 に記載のデッドロック事前検出プログラム。

【請求項 4】

前記第 3 手順は、前記処理ルートから、第 1 アクセスステップ、及び該第 1 アクセスステップの次の第 2 アクセスステップを取得し、

前記第 4 手順は、前記第 1 及び第 2 アクセスステップそれぞれによるデータベースへのアクセス順序が予め定められたアクセス順序か否かを判定し、

前記第 5 手順は、前記アクセス順序が予め定められたアクセス順序でないと判定された場合に、前記第 1 アクセスステップによるデータベースへのアクセスが、前記第 2 アクセスステップによるデータベースへのアクセスよりも先に行われている旨を報知する、
請求項 1 に記載のデッドロック事前検出プログラム。

【請求項 5】

多重アクセス記述と前記処理ルートを読み込ませる第 8 手順、

前記多重アクセス記述と処理ルートに基づいて、前記複数の処理ステップにより構成される業務ロジックを同時実行する場合に、デッドロックが発生する可能性の情報を生成する第 9 手順、

をさらに備える、

請求項 1 に記載のデッドロック事前検出プログラム。

**【書類名】 明細書****【発明の名称】** デッドロック事前検出プログラム**【技術分野】****【0001】**

本発明は、デッドロック検出対象のプログラムを実際に動作させることなく、事前にデッドロックの可能性を検出するための技術に関する。

【背景技術】**【0002】**

従来、デッドロック検出方法として、デッドロック検出の対象となる手続き（処理ステップ等）を実行、又はシミュレートして得られたデータアクセスシーケンスを分析することでデッドロックの検出を行う技術が提案されている。

【0003】

この従来のデッドロック検出方法の例について図17に示す。同図は、実行中のアプリケーションがデータベースに対してアクセスした履歴（図中SQL履歴）を履歴抽出器（図中SQL履歴抽出器）により記録し、このアクセス履歴に基づいて、デッドロック検出器が所定処理を実行することにより、デッドロック検出結果を出力することを示している（例えば、特許文献1、及び特許文献2等参照）。

【特許文献1】 特開2000-222228号公報**【特許文献2】** 特開平10-49389号公報**【発明の開示】****【発明が解決しようとする課題】****【0004】**

しかしながら、従来のデッドロック検出方法によれば、デッドロックを検出するには、デッドロック検出対象のプログラムを実際に実行する必要がある。このため、以下の問題点がある。

【0005】

（1）実際に動作させた部分についてしかデッドロックを検出できない。換言すれば、全てのケース（全ての処理ルート）についてデッドロックを検出したこと（網羅性）の証明が難しい。特に多重アクセス時のプログラム検証を行う場合、全ての可能性をチェックするのは事実上不可能に近い。

【0006】

（2）既にプログラムができてからチェックを行うため、デッドロック解消のためにプログラムを修正するためのコストが大きくなる（設計段階での修正の10倍～100倍程度）。

【0007】

（3）各業務ロジックを分担して開発する場合、担当者によってデータベースアクセス順序の実装にばらつきがあるためデッドロックを起こし得るが、プログラムができてからチェックする方法ではこの問題を解消できない。

【0008】

本発明の第1の課題は、デッドロック検出対象のプログラムを実際に動作させることなく、事前にデッドロックの可能性を検出することにある。また、本発明の第2の課題は、そのように検出されたデッドロック可能性回避のための方策等について利用者に対して報知することにより、デッドロック可能性回避のための方策等を採用することができるようにすることにある。

【課題を解決するための手段】**【0009】**

本発明は、上記の課題を解決するために次のように構成した。コンピュータに、複数のデータベースのうちのいずれかへのアクセスを伴うアクセスステップを含む複数の処理ステップにより構成される業務ロジック設計情報を読み込ませる第1手順、前記業務ロジック設計情報に基づいて、少なくとも2つのアクセスステップにより構成される処理ルート



を生成する第2手順、前記処理ルートから、第1アクセスステップ、及び第2アクセスステップを取得する第3手順、前記第1及び第2アクセスステップそれぞれによるデータベースへのアクセス順序が予め定められたアクセス順序か否かを判定する第4手順、前記アクセス順序が予め定められたアクセス順序でないと判定された場合に、予め定められたアクセス順序から逸脱した旨を報知する第5手順、を実行させるためのデッドロック事前検出プログラム。

【0010】

本発明によれば、業務システムを構成するプログラムを設計する段階（即ち業務ロジック設計書作成段階）で、デッドロックの可能性を検出することが可能となる。即ち、デッドロック検出対象のプログラムを実際に動作させることなく、事前にデッドロックの可能性を検出することが可能となる。

【0011】

このように、事前にデッドロックの可能性を検出することが可能となるため、デッドロック解消のためにプログラムを修正するためのコストが大きくなることもない。また、業務ロジックを分担して開発する場合、担当者によってデータベースアクセス順序の実装にばらつきが生ずることによるデッドロックの可能性を低減できる。

【0012】

また、上記デッドロック事前検出プログラムにおいては、例えば、前記業務ロジック設計情報は、前記アクセスステップ及び分岐条件ステップを含む複数の処理ステップにより構成され、前記第2手順は、前記業務ロジック設計情報に基づいて、少なくとも2つのアクセスステップにより構成される少なくとも2つの処理ルートを生成し、前記第3手順は、各処理ルートごとに、その処理ルートから、第1アクセスステップ、及び第2アクセスステップを取得し、前記第4手順は、各処理ルートごとに、前記第1及び第2アクセスステップによるデータベースへのアクセス順序が予め定められたアクセス順序か否かを判定する。

【0013】

このようにすれば、分岐条件ステップの各分岐先を経由することで複数の処理ルートが生成されたとしても、これら全処理ルートについて網羅的にデッドロックの可能性を検出することが可能となる。

【0014】

また、上記デッドロック事前検出プログラムにおいては、例えば、前記複数のデータベース相互間の対応関係を表す対応関係データを読み込ませる第6手順、前記対応関係データに基づいて、前記予め定められたアクセス順序を生成する第7手順、をさらに備える。

これは、予め定められたアクセス順序をどのようにして得るかについての一例を示したものである。利用者等により入力（設定）されたもののアクセス順序を、予め定められたアクセス順序としてもよい。

【0015】

このようにすれば、予め定められたアクセス順序を自動的に生成することが可能となる。

また、上記デッドロック事前検出プログラムにおいては、例えば、前記第3手順は、前記処理ルートから、第1アクセスステップ、及び該第1アクセスステップの次の第2アクセスステップを取得し、第4手順は、前記第1及び第2アクセスステップそれぞれによるデータベースへのアクセス順序が予め定められたアクセス順序か否かを判定し、前記第5手順は、前記アクセス順序が予め定められたアクセス順序でないと判定された場合に、前記第1アクセスステップによるデータベースへのアクセスが、前記第2アクセスステップによるデータベースへのアクセスよりも先に行われている旨を報知する。

【0016】

このようにすれば、記第1アクセスステップによるデータベースへのアクセスが前記第2アクセスステップによるデータベースへのアクセスよりも先に行われている旨、即ち、デッドロック可能性回避のための方策等について、利用者に対して報知されることになる。

。

【0017】

従って、この報知を受けた利用者は、そのデッドロック可能性回避のための方策等を探ることが可能となる。また、業務システムを構成する各業務ロジック、たとえば見積依頼伝票や購買依頼伝票の発行、承認といった業務ロジックにおける標準的なデータアクセス順序を決め、各業務ロジックにおいてデータアクセス順序を統一することも可能となる。

【0018】

また、上記デッドロック事前検出プログラムにおいては、例えば、多重アクセス記述と前記処理ルートを読み込ませる第8手順、前記多重アクセス記述と処理ルートに基づいて、前記複数の処理ステップにより構成される業務ロジックを同時実行する場合に、デッドロックが発生する可能性の情報を生成する第9手順、をさらに備える。

【0019】

このようにすれば、多重アクセス時におけるデッドロックの可能性を検出し、デッドロック回避の対策をとることができる。

また、本発明は、上記デッドロック事前検出プログラムを記録したコンピュータ読み取り可能な記録媒体として特定することもできる。

【発明の効果】

【0020】

デッドロック検出対象のプログラムを実際に動作させることなく、事前にデッドロックの可能性を検出することが可能となる。また、そのように検出されたデッドロック可能性回避のための方策等について利用者に対して報知することにより、デッドロック可能性回避のための方策等を探ることが可能となる。

【発明を実施するための最良の形態】

【0021】

以下、本発明の一実施形態であるプログラムが適用されるデッドロック防止システムについて図面を参照しながら説明する。

(デッドロック事前検出システムの概略システム構成)

図1は、本実施形態のデッドロック事前検出システムの概略システム構成について説明するための図である。図1に示すように、本実施形態のデッドロック事前検出システムは、処理シーケンス生成装置M1、標準アクセスシーケンス生成装置M2、デッドロック事前検出装置M3、多重アクセス時デッドロック事前検出装置M4、及びこれらの各装置M1～M4に読み込まれる各種データ（業務ロジック設計情報D11、データ仕様データ相関D21、及び多重アクセス操作系列記述D4等）が格納されたハードディスク装置等の記憶装置を備えている。これら各装置M1～M4は、例えば、パーソナルコンピュータ等の既存の情報処理端末が所定のプログラムを読み込み、これを実行することにより実現される。

(処理シーケンス生成装置)

処理シーケンス生成装置M1は、記憶装置から業務ロジック設計情報D11を読み込み、その業務ロジック設計情報D11に基づいて、後述の処理（図3及び図4参照）を実行する。これにより、処理シーケンス生成装置M1は、各ルートの処理シーケンス（処理ルート）を生成する。

(業務ロジック設計情報)

業務ロジック設計情報D11は、複数の処理ステップにより構成される。これら複数の処理ステップ全体により特定業務のロジック（例えば購買依頼伝票の一覧を取得するための業務ロジック）が構成される。処理ステップとしては、複数のデータベース（又は、テーブルともいう）のうちのいずれかへのアクセスを伴うアクセスステップ、分岐条件ステップ、開始ステップ、及び終了ステップ等がある。

【0022】

業務ロジック設計情報D11は、各種の方法で作成することが可能である。本実施形態では、複数の処理ステップを、情報処理端末が処理可能なスクリプト言語等の簡易言語で

記述する。なお、スクリプト言語で記述された業務ロジック設計情報 D 1 1 は、実行形式のプログラムにコード化することで実行可能となる。

【0023】

次に、業務ロジック設計情報 D 1 1 の具体例として、購買依頼伝票の一覧を取得するための業務ロジックを説明する。図 2 は、購買依頼伝票の一覧を取得するための業務ロジックについて説明するための図である。

【0024】

この業務ロジック設計情報 D 1 1 は、開始ステップ S 1、データベースへのアクセスステップ S 2、S 4、S 5、S 7、フラグ設定ステップ S 3、分岐条件ステップ S 6、及び終了ステップ S 8 等の複数の処理ステップ S 1～S 8 により構成される。

【0025】

次に、各ステップの処理内容の詳細、及び各ステップ（例えば動作や動作対象等により表される）をスクリプト言語で記述した例について説明する。

開始ステップ S 1 では、処理内容が「購買伝票の検索開始」、記述例が「start」となる。アクセスステップ S 2 では、処理内容が「承認フロー検索（業務ロジック実行者が所属する組織における承認フロー情報を、承認フローデータベースから検索（データベースに対しては参照系のアクセス））」、記述例が「read承認フローデータベース」となる。この記述例は、承認フローデータベースに対するアクセスが発生することを表している。

【0026】

フラグ設定ステップ S 3 では、処理内容が「承認フローが S 2 の検索で見つかった場合、承認フローフラグを ON に設定」、記述例が「set apr_flow-flag」となる。この記述例は、apr_flow-flag がセットされることを表している。

【0027】

アクセスステップ S 4 では、処理内容が「購買依頼伝票の検索（データベースに対しては参照系のアクセス）」、記述例が「read購買伝票本体データベース」となる。この記述例は、購買伝票本体データベースに対するアクセスが発生することを表している。

【0028】

アクセスステップ S 5 では、処理内容が「検索の結果見つかった各伝票の品目情報検索（データベースに対しては参照系の検索）」、記述例が「read品目情報データベース」となる。この記述例は、品目情報データベースに対するアクセスが発生することを表している。

【0029】

分岐条件ステップ S 6 では、処理内容が「承認フローフラグを参照し、フラグが ON か否かの判定」、記述例が「check apr_flow-flag if true S7 if false S8」となる。この記述例は、check apr_flow-flag が真であれば S 7 を実行し、一方、check apr_flow-flag が偽であれば S 8 を実行することを表している。

【0030】

アクセスステップ S 7 では、処理内容が「承認フローフラグ（check apr_flow-flag）が ON の場合は、S 4 で検索された伝票に対する承認履歴情報データベースを検索（データベースに対しては参照系のアクセス）」、記述例が「read承認履歴情報データベース」となる。この記述例は、承認履歴情報データベースに対するアクセスが発生することを示している。

【0031】

最終ステップ S 8 では、処理内容が「検索結果（購買依頼伝票、品目情報、承認フローフラグが ON の場合は承認履歴情報も含む）を返却」、記述例が「end」となる。

（処理ルート情報を生成する処理）

次に、処理シーケンス生成装置 M 1 が、業務ロジック設計情報 D 1 1 に基づいて、処理ルート情報（各ルートの処理シーケンス）を生成する処理について説明する。図 3 は、処理ルート情報を生成する処理について説明するためのフローチャートである。処理手順は以下のとおりである。

【0032】

処理シーケンス生成装置M1は、業務ロジック設計情報D11を構成する処理ステップS1～S8に対して以下の処理を実行する。

処理シーケンス生成装置M1は、業務ロジック設計情報D11から、一つずつ処理ステップを取り出し（S100:Yes、S101）、その取り出した処理ステップが条件分岐ステップか否かを判定する（S102）。そして、処理シーケンス生成装置M1は、その取り出した処理ステップが条件分岐ステップでなければ（S102:No）、その取り出した処理ステップを現在の処理ルート情報R1に格納（追加）する（S103）。

【0033】

図2を参照すると、業務ロジックの最初の処理ステップS1～S5までは、条件分岐ステップではない。このため、処理シーケンス生成装置M1は、最初の処理ステップS1～S5を、現在の処理ルート情報R1に格納（追加）する（S100～S103）。

【0034】

次に、処理シーケンス生成装置M1は、業務ロジック設計情報D11から処理ステップS6を取り出す（S100:Yes、S101）。この処理ステップS6は条件分岐ステップ(if文)であるので（S102:Yes）、処理シーケンス生成装置M1は、現在の処理ルート情報R1のコピーR2（処理ステップS1～S5が格納されている）を生成する（S104）。

【0035】

そして、処理シーケンス生成装置M1は、処理ルート情報R1に対しては条件が成立する場合（図2ではS6:Yes）の処理ステップ（図2ではS7）を格納（追加）する処理を実行する（S105）。これにより、処理ルート情報R1の格納内容は、処理ステップS1～S5、及びS7となる。一方、処理ルート情報R2に対しては条件が成立しない場合（図2ではS6:No）の処理ステップ（図2ではS8）を格納（追加）する処理を実行する（S106）。これにより、処理ルート情報R2の格納内容は、処理ステップS1～S5、及びS8となる。これは、一方のルート（以下ルート2という）の処理シーケンスを表す。

【0036】

次に、処理シーケンス生成装置M1は、業務ロジック設計情報D11から処理ステップS8を取り出す（S100:Yes、S101）。この処理ステップS8は条件分岐ステップでないため、処理シーケンス生成装置M1は、この処理ステップS8を現在の処理ルート情報R1に追加する（S103）。これにより、処理ルート情報R1の格納内容は、処理ステップS1～S5、S7、及びS8となる。これは、他方のルート（以下ルート1という）の処理シーケンスを表す。

【0037】

上記各ルート1、2の処理シーケンスそれぞれに対しては、図4に示すシーケンス抽出処理が実行される。

図4は、各処理ルート（各ルート1、2の処理シーケンス）に対するデータアクセスシーケンス（図6参照）を抽出する処理について説明するためのフローチャートである。処理手順は以下のとおりである。

【0038】

処理シーケンス生成装置M1は、各処理ルートに対して以下の処理を実行する。

処理シーケンス生成装置M1は、アクセスシーケンス（データベースアクセスシーケンス）として空のシーケンスを生成する。そして、処理シーケンス生成装置M1、処理ルート1から、処理ステップを一つずつ取り出し（S200:Yes、S201）、その取り出した処理ステップを解析する。具体的には、その処理ステップの復帰値、手続き名、及び引数等（又は、readやwrite等、データベースに対する操作や、承認フローデータベース等、その操作対象データベース）を解析する。

【0039】

その解析の結果、処理ステップがデータベースへのアクセスの場合（S203:Yes

)、処理シーケンス生成装置M1は、アクセスシーケンスに、そのアクセス対象データベース(テーブル)とそのアクセス内容(Read/Writeの区別、又はR/Wモードの区別)の対を格納(追加)する(S204)。例えば、処理ルート1については、図5に示すように、このルートを構成する処理ステップのうちS2、S4、及びS5がデータベースへのアクセスを伴う。一方、ルート2については、このルートを構成する処理ステップのうちS2、S4、S5、及びS7がデータベースへのアクセスを伴う。

【0040】

従って、処理シーケンス生成装置M1は、ルート1については、アクセスシーケンスに、アクセス対象データベースとその操作内容の対として、承認フローデータベース(Read)→購買伝票本体データベース(Read)→品目情報データベース(Read)の順に格納(追加)する(S203:Yes、S204)。また、処理ルート2については、別のアクセスシーケンスに、アクセス対象データベースとそのアクセス内容の対として、承認フローデータベース(Read)→購買伝票本体データベース(Read)→品目情報データベース(Read)→承認履歴情報データベース(Read)の順に格納(追加)する(S203:Yes、S204)。

【0041】

一方、処理ルート1、2のいずれについても、図5に示すように、これらのルートを構成する処理ステップのうちS1、S3、S6、及びS8は、データベースへのアクセスを伴わない(S203)。この場合、他手続きの呼び出しもないため(S205:No)、処理シーケンス生成装置M1は、S200以下の処理を繰り返す。

【0042】

以上述べたように、各処理ルート1、2のアクセスシーケンスには、アクセス対象データベースとそのアクセス内容の対が順に格納(追加)される。従って、このアクセスシーケンスを参照することにより、各処理ルート1、2のアクセス対象データベース、及びそれら各データベースへのアクセス順序を把握することが可能となっている。

【0043】

一方、S202での解析の結果、処理ステップがデータベースへのアクセスでない場合(S203:No)、処理シーケンス生成装置M1は、その処理ステップが手続き呼び出しか否かを判定する(S205)。その判定の結果、その処理ステップが手続き呼び出しの場合は(S205:Yes)、処理シーケンス生成装置M1は、呼び出す手続きについてこの手続きを再帰的に実行し、得られたアクセスシーケンスを追加する(S206~S208)。なお、呼び出しの可能性がある各手続きについて、その手続きを呼び出した際に実行されるアクセスシーケンスを予め用意しておき、手続き呼び出しに対して直接対応するアクセスシーケンスを取得して追加する方法を取ることでもできる。

【0044】

なお、業務ロジック設計情報から取り出した処理ステップが条件分岐ステップであるか否かについては、S102でその取り出した処理ステップがif文であるか否かにより判定するように説明したが、これに限定されない。例えば、同処理ステップがswitch文(又はその他の条件分岐命令)であるか否かにより判定するようにしてもよい。

【0045】

なお、If文による条件判定がN回ある場合には、2のN乗とおりの処理ルート情報が生成される。多値分岐がある場合は、各分岐における選択肢の数を掛け合わせた数の処理ルート情報が生成される。たとえばif文が4回、値4通りの多値分岐と値5通りの多値分岐が現れる場合は $2^4 \times 4 \times 5 = 320$ 通りの処理ルート情報が生成される。

【0046】

なお、図6は、図2に示した業務ロジック設計情報から生成される処理ルート情報のうちの、一つの処理ルートにおけるデータアクセスシーケンス図である。図6では図5に示した2つの処理ルートのうち、承認フローフラグがOFFの場合のデータアクセスシーケンス図である。メインの業務ロジックは図6の左端にある「購買伝票一覧取得」である。

【0047】

図6においては、承認ルートを参照するときには承認フローAPI、承認フロー本体操作

の2つの手続きを経由して承認ルートデータベースを参照する。購買伝票本体を参照するときには購買伝票API、購買伝票本体操作の2つの手続きを経由して購買伝票本体データベースを参照する。品目情報を参照するときには購買伝票API、品目情報操作の2つの手続きを経由して品目情報のデータベースをアクセスする。

(標準アクセスシーケンス生成装置)

標準アクセスシーケンス生成装置M2は、記憶装置からデータ仕様データ相関D21を読み込み、そのデータ仕様データ相関D21に基づいて、後述の処理(図10参照)を実行することにより、標準アクセスシーケンスD22を生成する。

(データ仕様データ相関)

図7は、データ相関図の例を説明するための図である。ここでは、説明の便宜上、図2に示した各データベースとは異なる複数のデータベースを取り上げて説明する。このデータ相関図は、複数のデータベースに対する標準的なアクセス順序を決定するために、複数のデータベース相互間の対応関係を可視的に表したものである。なお、この対応関係を表す対応関係データの例については図8、及び図9に示している(詳細については後述)。

【0048】

図7では複数のデータベースとして8つのテーブル、即ち購買伝票本体テーブル、発注品目情報テーブル、購買申請者情報テーブル、販売担当者情報テーブル、商品情報テーブル、価格情報テーブル、購買企業情報テーブル、及び販売企業情報テーブルを示している。

【0049】

これらテーブル相互間の対応関係は、2つのテーブルを結ぶ矢印によって示される。即ち、矢印の先端で指されたカラムが、その矢印の基端のカラムよりも後にアクセスするカラムになる。例えば、購買伝票本体テーブルのslipID(Primary Key)が基端、かつ、発注品目情報テーブルのslipIDが先端の矢印により、両テーブルは結ばれている。これは、矢印の先端で指された発注品目情報テーブルのslipIDが、その矢印の基端の購買伝票本体テーブルのslipID(Primary Key)よりも後にアクセスされることを意味する。即ち、購買伝票本体と発注品目情報では、購買伝票本体のslipID(Primary Key)を取得し、発注品目情報テーブルに対して矢印の先端が指すslipIDが購買伝票本体のslipIDと同じ、という条件での検索を行う。

【0050】

また、テーブル相互間の対応関係としては、次のようなものもある。図7を参照すると、発注品目情報テーブルのカラムslipidは購買伝票本体テーブルのカラムslipid(プライマリキー)の値を持ち、発注品目情報と購買依頼伝票の対応関係はN:1となっている(1件の購買依頼伝票に対してN件の発注品目情報が対応する)。同様に、価格情報テーブルのproductIdは商品情報のプライマリキーidの値であり、価格情報と商品情報の対応関係はN:1となっている(1件の商品情報に対してN件の価格情報が対応)。また、発注品目情報のproductIdは商品情報のidの値で、発注品目情報と商品情報の対応は1:1(1件の発注品目情報に対し、1件の商品情報が対応)となっている。これらN:1や1:1も、テーブル相互間の対応関係を表すためのものである。

【0051】

図7に示したテーブル相互間の対応関係は、これら対応関係を表す対応関係データとして、情報処理端末により保持される。図8は、テーブル相互間の対応関係を表す対応関係データ(データ仕様データ相関)の例について説明するための図である。

【0052】

ここでは、説明の便宜上、図7に示した複数のテーブルとは異なるテーブル(CT_TUSPARテーブル)を取り上げて説明する。なお、実際には、図7に示した各テーブルそれぞれに対して、図8に示したテーブル相互間の対応関係を表す対応関係データに相当するデータが存在する。図9は、図8の対応関係データをXML言語で記述した例である。なお、図8の対応関係データについては、XMLの他、JavaやC++等の様々の言語で記述することが可能である。

【0053】

図8は、利用者情報を格納するCT_TUSRPARテーブルに対する対応関係データ（データ仕様）である。このデータ仕様は、CT_TUSRPARテーブルとCT_TCMPCOMテーブル間の対応関係を表す対応関係データを含む。この対応関係データとしては、図8中段部分の"CORRESPONDS CT_TCMPCOM.COMID (改行) TYPE=1:1 (改行) ACCESS-SQUESNCE USRCOMP->CT_TCMPCOM.COMID (改行)"が相当する。これは、矢印の先端で指されたCT_TCMPCOMテーブルのカラムCOMIDが、その矢印の基端のCT_TUSRPARテーブルのカラムUSRCOMPよりも後にアクセスされることを意味する。また、"TYPE=1:1"の部分が、CT_TCMPCOMとCT_TCMPCOMとが1:1の関係にあることを表している。

【0054】

なお、CT_TUSRPARテーブルは、テーブル名がCT_TUSRPAR、レコード数の想定値が3000レコードで、カラムとして利用者管理番号USERNUM、利用者名USERNAME、パスワードUSRPWD、利用者の所属企業USRCOMP、利用者所属事業所USROFFICE、利用者所属組織USBELONG、利用者メールアドレスMAILADDR、利用者の電話番号PHONENUMBER、利用者のFAX番号FAXNUMBERを持ち、レコードを一意に決めるプライマリキーはUSERNUMとなっている。

（標準アクセス順序を生成する処理）

次に、標準アクセスシーケンス生成装置M2が、標準アクセスシーケンスD2を生成する処理について説明する。図10は、標準データアクセスシーケンスを生成する処理について説明するためのフローチャートである。処理手順は以下のとおりである。

【0055】

標準アクセスシーケンス生成装置M2は、図7に示した各テーブルTAごとに、そのテーブルTAとそのテーブルTBとは別の各テーブルTBとの関係で、以下の処理を実行する。

【0056】

標準アクセスシーケンス生成装置M2は、図7に示した複数のテーブルの中から、処理対象のテーブルTAを一つ選択し、そのテーブルTAのカラムのうち、カラムの値が各テーブルTBのカラムの値になっているカラムの数N1、及びテーブルTBのカラムのうち、カラムの値がテーブルTAのカラムの値になっているカラムの数N2を求める（S300、S301）。

【0057】

次に、標準アクセスシーケンス生成装置M2は、N1とN2を比較し、 $N1 > N2$ か否かを判定する（S302）。その結果、N1がN2より大であれば（S302:Yes）、テーブルTAをテーブルTBよりも前のアクセス順序にする（S303）。例えば、リスト上の配置順序を変える。又はテーブルTAに対するアクセス優先度を増やし、テーブルTBに対するアクセス優先度を減らす。

【0058】

一方、S302の判定の結果、 $N1 = N2$ の場合は（S302:No、S304:Yes）、標準アクセスシーケンス生成装置M2は、テーブルTAの平均レコード数（図9に示したデータ仕様のヘッダ部分から得られる）とテーブルTBの平均レコード数とを比較し、テーブルTAの平均レコード数 \leq テーブルTBの平均レコード数か否かを判定する（S305）。

【0059】

その結果、テーブルTAの平均レコード数 \leq テーブルTBの平均レコード数であれば（S305:Yes）、平均レコード数の少ない方のテーブルTAを、平均レコード数の多いテーブルTBより前のアクセス順序にする（S306）。例えば、リスト上の配置順序を変える。又は、平均レコード数の少ないテーブルTAの優先度を上げ、平均レコード数の多いテーブルTBの優先度を上げる。

【0060】

一方、S304の判定の結果、N1がN2より小であれば（S302:No、S304:No）、テーブルTBをテーブルTAより前のアクセス順序にする。例えばリスト上の配置順序

を変える。又は、テーブルTBに対するアクセス優先度を増やし、TAに対するアクセス優先度を減らす。

【0061】

なお、S304の判定の結果、N1とN2が等しく（S302：No、S304：Yes）、S305の判定の結果、平均レコード数も等しい場合には、例えば、テーブルTAのカラム値がテーブルTBのプライマリキーの値である場合にテーブルTAをテーブルTBより前のアクセス順序にする。又は、テーブルTAに対するアクセス優先度を増やし、テーブルTBに対するアクセス優先度を減らす。

【0062】

標準アクセスシーケンス生成装置M2は、図7に示した複数のテーブルの中から、選択できるテーブルがなくなるまで、処理対象の次のテーブルTAを一つ選択し、上記S300～S307の処理を繰り返す。

【0063】

以上の処理により、最終的に得られるテーブルの順序（又は優先度の高い順にテーブルを並べたもの）が、標準的なテーブルのアクセス順序になる。

図11は、標準データアクセスシーケンスの例を説明するための図である。図11は、図7に示した複数のテーブルへの標準的なアクセス順序が、購買依頼伝票テーブル、品目情報テーブル、購買申請者情報テーブル、販売担当者情報テーブル、商品情報テーブル、価格情報テーブル、購買企業情報テーブル、販売企業情報テーブルであることを示している。

（デッドロック事前検出装置）

デッドロック事前検出装置M3は、各ルートの処理シーケンス、及び標準アクセスシーケンスに基づいて、後述の処理（図12参照）を実行することにより、標準アクセス系列からの逸脱情報D3を生成する。

（逸脱情報生成処理）

次に、デッドロック事前検出装置M3が、各ルートの処理シーケンスD12、及び標準アクセスシーケンスD22に基づいて、標準アクセス系列からの逸脱情報D3を生成する処理について説明する。図12は、図6に示した業務ロジックのあるルートにおけるアクセスシーケンスとの逸脱情報を抽出する処理について説明するためのフローチャートである。処理手順は以下のとおりである。

【0064】

デッドロック事前検出装置M3は、各処理ルート（各ルートのアクセスシーケンス）に対して以下の処理を実行する。

デッドロック事前検出装置M3は、ある一つの処理ルートRに対して既に出現済のデータアクセスを格納する格納庫Sを空にする（S400）。デッドロック事前検出装置M3は、その処理ルートRの中のデータアクセスArを一つづつ順に取り出す（S401：No、S402）。なお、全てのデータアクセスArを取り出した場合は処理を終了する（S401：Yes）。

【0065】

次に、デッドロック事前検出装置M3は、格納庫Sに、既に取り出し済のデータアクセスArがあるか否かを判定する（S403）。その結果、データアクセスArがない場合（S403：No）には、格納庫SにデータアクセスArを格納（追加）し、S401に戻る（S404）。

【0066】

一方、S403の判定の結果、既に取り出し済のデータアクセスArがある場合には（S403：Yes）、格納庫Sからチェック対象のデータアクセスAsを読み出して（S406）、標準のデータアクセス順序を参照して、データアクセスArがデータアクセスAsより先か否かを判定する（S407）。その結果、データアクセスArがデータアクセスAsより先であれば（S407：Yes）、デッドロック事前検出装置M3は、データアクセスArがデータアクセスAsより先に実行されているという警告を作成する（S408）。

【0067】

そして、デッドロック事前検出装置は、その旨の警告が既に出力済か否かを判定（例えば後述のフラグに基づく判定）し（S409）、その結果、出力済でなければ（S409：No）その旨の警告を出力して（S410）、その警告は出力済であることを示すフラグをセットする（S411）。

【0068】

なお、警告の出力形態としては各種のものが考えられるが、ここでは、図13に示すように、ログ（例えばログファイル）として出力（報知）される。以上のS406～S411の処理は、格納庫Sに格納されている全てのデータアクセスAsに対して行われる（S405）。全てのデータアクセスAsに対する処理が完了すると（S405：Yes）、デッドロック事前検出装置M3は、格納庫SにデータアクセスArを格納（追加）し、S401に戻る（S404）。

【0069】

以上の処理により、図13下段に示した逸脱情報が得られる。業務システムの設計者等は、この逸脱情報を参照することで、標準のデータアクセス順序に従っていない業務ロジックのデータアクセス順序を修正して、業務ロジックにおけるデータアクセス順序が標準のデータアクセス順序になるようにすることが可能となる。

【0070】

図13は、標準データアクセスシーケンスに対する、ある業務ロジックのあるルートにおけるデータアクセスシーケンスの逸脱情報の例である。図13においては標準的なテーブルのアクセス順序が、(1)購買伝票本体、(2)品目情報、(3)商品情報、(4)価格情報、(5)承認フロー情報、(6)承認履歴情報、(7)利用者情報、(8)企業情報の順になっている。

【0071】

これに対して業務ロジック01050「購買依頼伝票の承認」の処理ルート01050-0001「利用者の種別が購買管理者の場合の処理ルート」では、(1)承認フロー取得(Read)、(2)購買依頼伝票検索(Read)、(3)発注品目情報検索(Read)、(4)商品情報取得(Read)、(5)価格情報検索(Read)、(6)承認履歴情報検索(Read)、(7)購買依頼伝票更新(Write)、(8)承認履歴情報更新(Write)の順になっている。

【0072】

標準的なテーブルのアクセス順序と比較すると、承認フロー情報が、購買伝票情報、品目情報、商品情報、価格情報の4つのテーブルより前にアクセスされるようになっているところが異なる。従って警告では承認フロー情報と、これら4つのテーブルのアクセス順序が標準のテーブルアクセス順序と異なることに対してそれぞれ警告を出している。

（多重アクセス時デッドロック事前検出装置）

多重アクセス時デッドロック事前検出装置M4は、各ルートの処理シーケンスD12、及び多重アクセス操作系列記述D11に基づいて、後述の処理（図16）を実行することにより、多重アクセスデッドロック可能性情報D4を生成する。

（多重アクセス記述）

図14は、多重アクセス記述、即ち業務ロジックの実行順序、及び同時実行される業務ロジックの実行系列を表現する情報の例である。

【0073】

図14において、操作系列T1は以下の業務ロジックを以下の順序で実行する。(1)販売支援でログインする。(2)見積・受注メニューを選択する。(3)見積回答を選択する。(4)見積伝票一覧表示を選択する。(5)一覧表示の中から1つ、見積依頼伝票を選択する。(6)選択した見積依頼伝票に対して見積回答を実行する。(7)販売支援からログアウトする。

【0074】

以上にあげた業務ロジックのうち、実際にデータベースに対してアクセスを行うのは(1)、(4)、(5)、(6)、(7)である。(2)、(3)は画面遷移のみの実行で、データベースに対してアクセスは行わない。従って、これから述べるデッドロックチェックでは(2)、(3)はチェックの対象外となる。

【0075】

また、(4)の見積伝票一覧表示では、以下の4通りのルートが存在する。

ルートA001：利用者が特定の購買企業を担当し、任意のカテゴリの商品を取り扱う。データアクセス順序は、販売担当情報→見積依頼伝票→見積依頼品目情報の順である。ルートA002：利用者が全購買企業を担当し、任意のカテゴリの商品を取り扱う。

【0076】

データアクセス順序は、見積依頼伝票→見積依頼品目情報の順である。

ルートA003：利用者が特定の購買企業を担当し、特定のカテゴリの商品を取り扱う。データアクセス順序は、販売担当情報→見積依頼伝票→見積依頼品目情報→商品情報の順である。

【0077】

ルートA004：利用者が全購買企業を担当し、特定のカテゴリの商品を取り扱う。データアクセス順序は、販売担当情報→見積依頼伝票→見積依頼品目情報→商品情報の順である。

【0078】

図15は、図14の多重アクセス記述と図4、5のアクセスシーケンス情報から検出されるデッドロック可能性の例である。図15においては操作系列T1の中の業務ロジックL1において、処理ルートR1が存在し、テーブルをTable1(Read)→Table2(Read/Write)→Table3(Write)→Table4(Read)の順にアクセスする。一方、操作系列T2の中に業務ロジックL2において、処理ルートR2が存在し、テーブルをTable1(Read)→Table3(Write)→Table2(Write)→Table4(Read)の順にアクセスする。ここで、R1とR2が同時に実行される場合、Table2とTable3においてデッドロック発生の可能性が出る。

【0079】

図16は、図14の多重アクセス記述と、図5、6のアクセスシーケンス情報から図15に示す多重アクセス時のデッドロックを検出するアルゴリズムである。処理手順としては、全ての操作系列T1、T2の組み合わせに対して以下を実行する。

- (1) T1の各業務ロジックL1、T2の各業務ロジックL2に対して以下を実行する。
- (2) L1の各処理ルートR1、L2の各処理ルートR2に対して以下を実行する。
- (3) R1に出現するデータアクセスから任意の2個を選んだ組み合わせC1を生成する。
- (4) R2に出現するデータアクセスから任意の2個を選んだ組み合わせC2を生成する。
- (5) C1の各データアクセスの対P1に対して、以下を実行する。
- (6) C2の各データアクセスの対から、P1と同じデータアクセスを持つデータアクセスの対P2を取り出す。
- (7) P1とP2においてデータアクセスの順が逆になっていたら、R1とR2の間でP1、P2に含まれるデータアクセスでデッドロックの可能性があると判断し、デッドロックの可能性を警告するメッセージを出力する（既に出力済みかどうかをチェックして、出力済みでなければ出力する）。

【0080】

全ての組み合わせに対してチェックするコストが大きい場合は、たとえば以下のようにしてチェックする操作系列を絞り込むことができる。

(A)各操作系列(または業務ロジック、業務ロジックの中の処理ルート)に対して発生頻度情報(単位時間あたりの発生数)を持たせる。発生頻度が一定の閾値以下であればその操作系列(または、業務ロジック、業務ロジックの中の処理ルート)はチェックの対象からはずす。

【0081】

(B)発生頻度情報がない場合、データテーブルに対してデッドロックのチェックを行う/行わないのフラグ情報を設定し、データアクセスの対に対してデッドロックのチェックを行うときに、対象となるテーブルがいずれもデッドロックを行うフラグがONになっている場合のみチェックを行う。

【0082】

以上述べたように、本実施形態のデッドロック防止システム（プログラム）によれば、次の効果を奏する。

(1) 業務システムの保守性向上 データアクセス処理系列を、全ての業務ロジックで遵守することにより業務システムを構成する業務ロジックの作りを統一することができる。業務ロジックの作りが統一されることにより、業務ロジックを記述するプログラムの見通しが良くなる。すなわち業務システム全体としての保守性が向上する。

【0083】

(2) 業務システムの品質確保 多重アクセス環境下で使用される業務システムにおいて、デッドロックの可能性を設計段階で網羅的に検出し、設計段階で業務システム全体にわたって、発生のある可能性があるデッドロックをすべた洗い出し、洗い出したデッドロックに対する防止策をとることができる。したがって、設計段階でデッドロック防止に関する品質の作りこみを可能にする。

【0084】

(3) 業務システム開発期間の短縮 テスト段階でデッドロックを逐一見つけて対策をとる方式にくらべて作業の手戻りを少なくすることができる。このため、設計から開発、テストを含めた業務システムの開発期間を2/3～1/2近くまで短縮することができる。

【0085】

本発明は、以下の分野にも適用可能である。

複数のデータ集合体から構成されるデータ格納庫を利用して業務処理を実行する装置において、複数の業務処理が同時に同じデータをアクセスすることによる資源競合(デッドロック)を運用前の設計段階で事前に検出するために適用される。

【0086】

データ格納庫の例としては、関係データベース(データ集合体がテーブル、各データがレコード)やオブジェクト指向データベース(データ集合体がオブジェクトの集まり、各データがオブジェクトと呼ばれるデータ構造)が考えられる。装置の例としては、電子商取引システムや会計システム、取引情報管理システム、電子カタログ管理システムなどが考えられる。

【0087】

電子商取引システムの場合、データ集合体としてたとえば利用者情報、企業情報、見積依頼伝票、購買依頼伝票、購入実績情報、商品情報、価格情報などのテーブルを持つ。これらのデータをアクセスする業務ロジックとしては見積依頼や購買依頼伝票の発行、見積依頼や購買依頼伝票に対する承認、見積回答や受注回答などが挙げられる。

【0088】

本発明は、その精神または主要な特徴から逸脱することなく、他の様々な形で実施することができる。このため、上記の実施形態は、あらゆる点で単なる例示にすぎず、限定的に解釈されるものではない。

【産業上の利用可能性】

【0089】

本発明によれば、デッドロック検出対象のプログラムを実際に動作させることなく、事前にデッドロックの可能性を検出することが可能となる。また、そのように検出されたデッドロック可能性回避のための方策等について利用者に対して報知することにより、デッドロック可能性回避のための方策等を採用することが可能となる。

【図面の簡単な説明】

【0090】

【図1】 本実施形態のデッドロック事前検出システム（デッドロック事前検出プログラムが適用される）の概略システム構成について説明するための図である。

【図2】 購買依頼伝票の一覧を取得するための業務ロジックについて説明するための図である。

【図3】 処理ルート情報を生成する処理について説明するためのフローチャートである。

- 【図 4】シーケンス抽出処理について説明するためのフローチャートである。
- 【図 5】処理ルート情報の例である。
- 【図 6】データアクセスシーケンス図の例である。
- 【図 7】データ相関図の例である。
- 【図 8】データ仕様の記述例である。
- 【図 9】データ仕様の記述例である（XML 表記）。
- 【図 10】標準データアクセスシーケンスを生成する処理について説明するためのフローチャートである。
- 【図 11】標準データアクセスシーケンスの例である。
- 【図 12】逸脱情報を生成する処理について説明するためのフローチャートである。
- 【図 13】逸脱情報等の例である。
- 【図 14】多重アクセスのための操作系列記述の例である。
- 【図 15】図 14 の多重アクセス記述と図 4、5 のアクセスシーケンス情報から検出されるデッドロック可能性の例である。
- 【図 16】多重アクセス時のデッドロックを検出するアルゴリズムである。
- 【図 17】従来のデッドロック検出方法について説明するための図である。

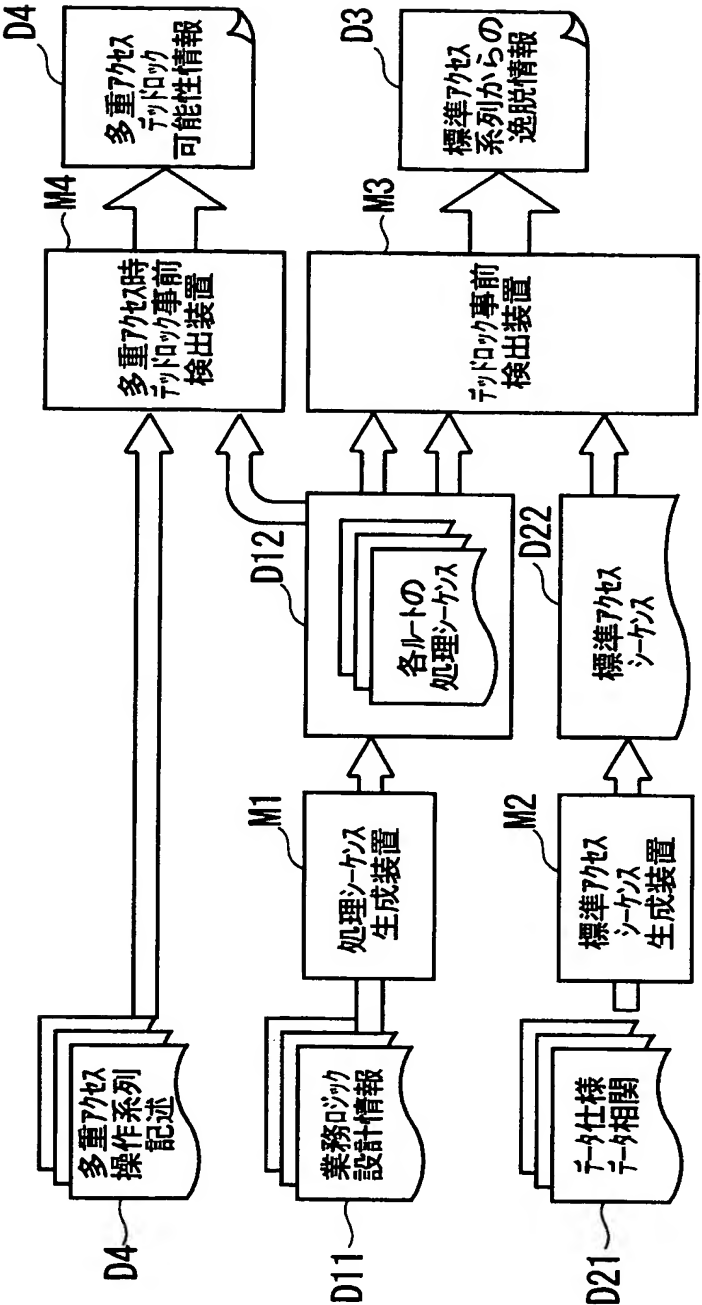
【符号の説明】

【0091】

- M 1 処理シーケンス生成装置
- M 2 標準アクセスシーケンス生成装置
- M 3 デッドロック事前検出装置
- M 4 多重アクセス時デッドロック事前検出装置
- D 1 1 業務ロジック設計情報
- D 1 2 処理シーケンス
- D 2 1 データ仕様データ相関
- D 2 2 標準アクセスシーケンス
- D 3 逸脱情報
- D 4 多重アクセスデッドロック可能性情報

【書類名】 図面
【図 1】

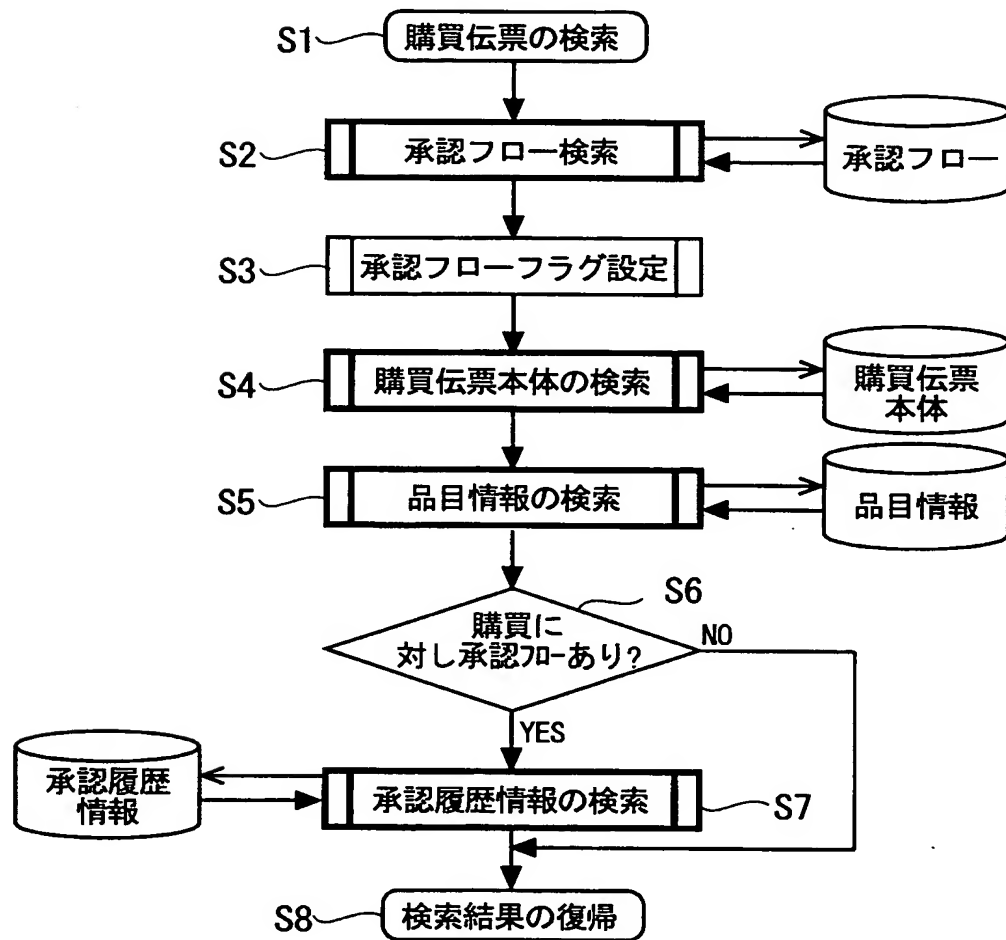
本実施形態のデッドロック事前検出システムの概略システム構成



【図 2】

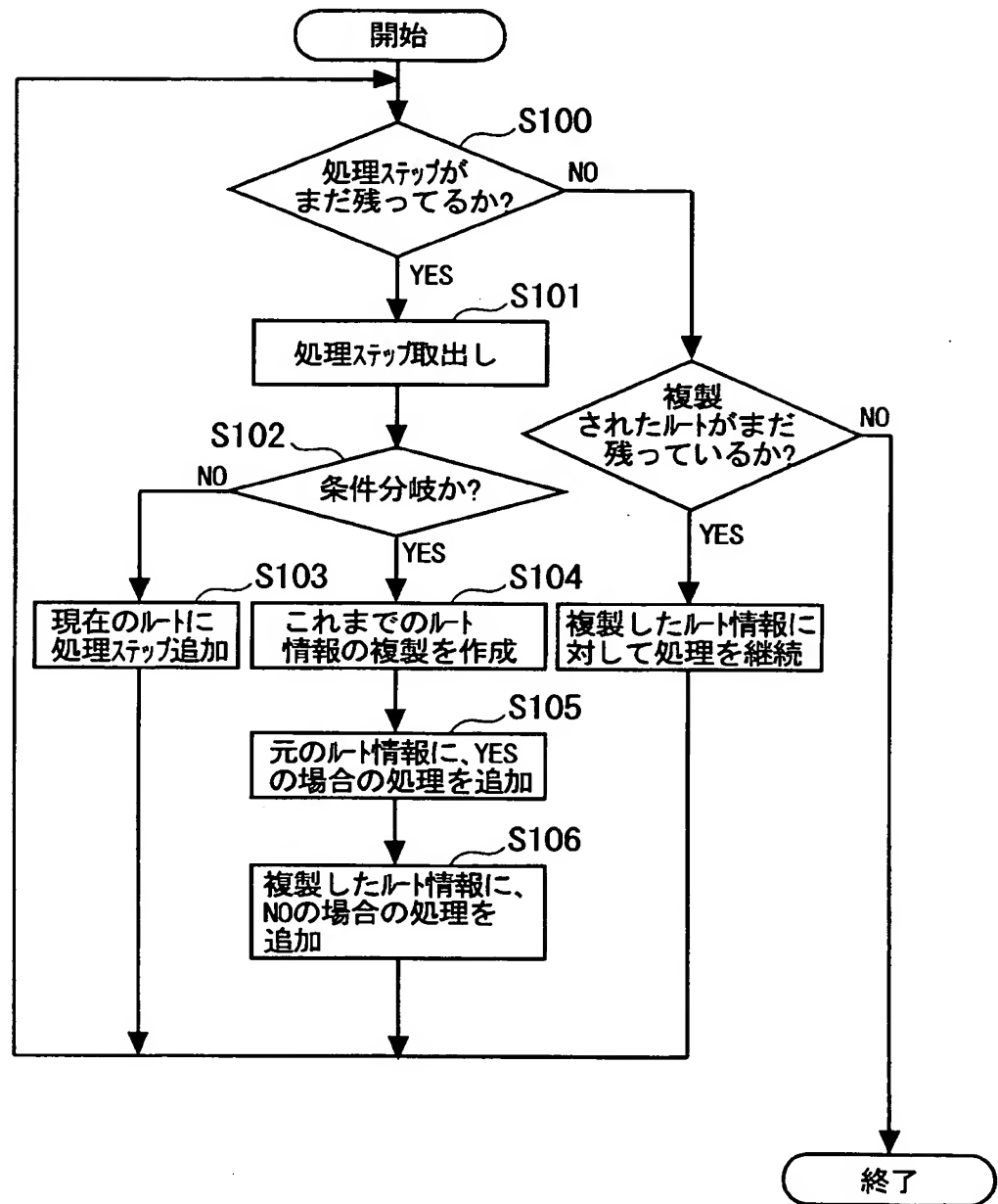
業務ロジック設計情報の例

(購買伝票検索と一覧取得の例を以下に挙げる)



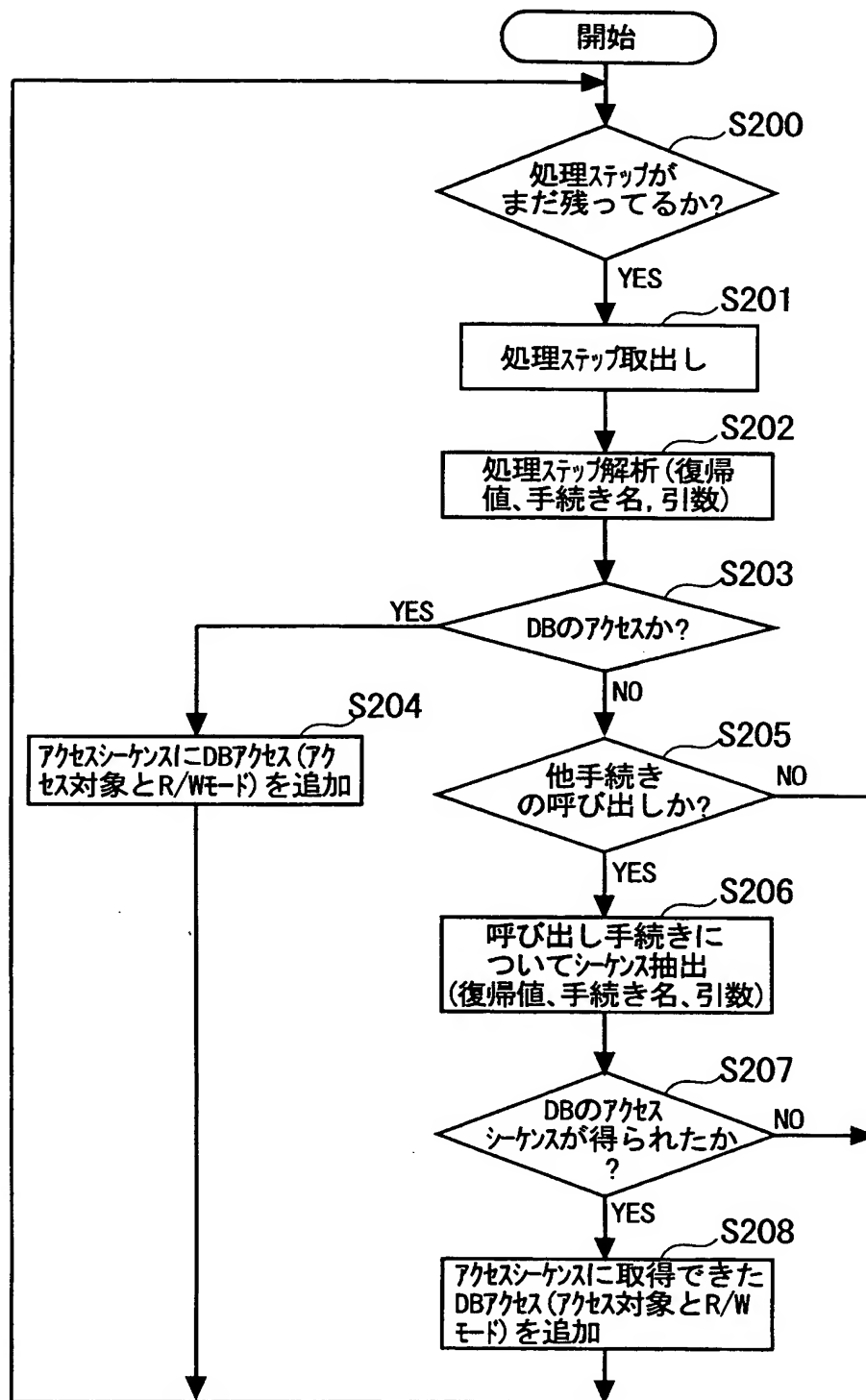
【図 3】

処理ルート情報を生成する処理
(設計情報からルート情報抽出)

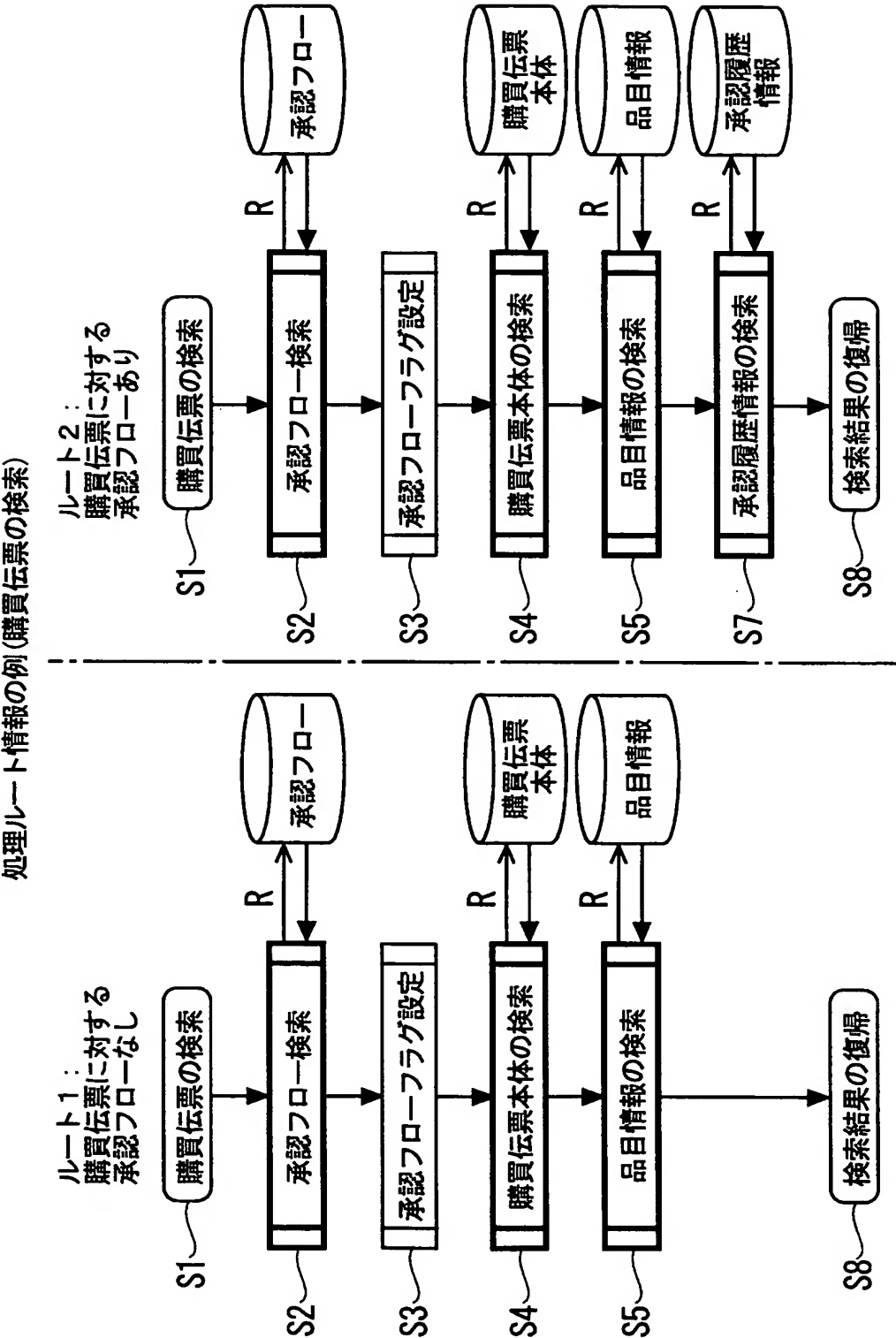


【図 4】

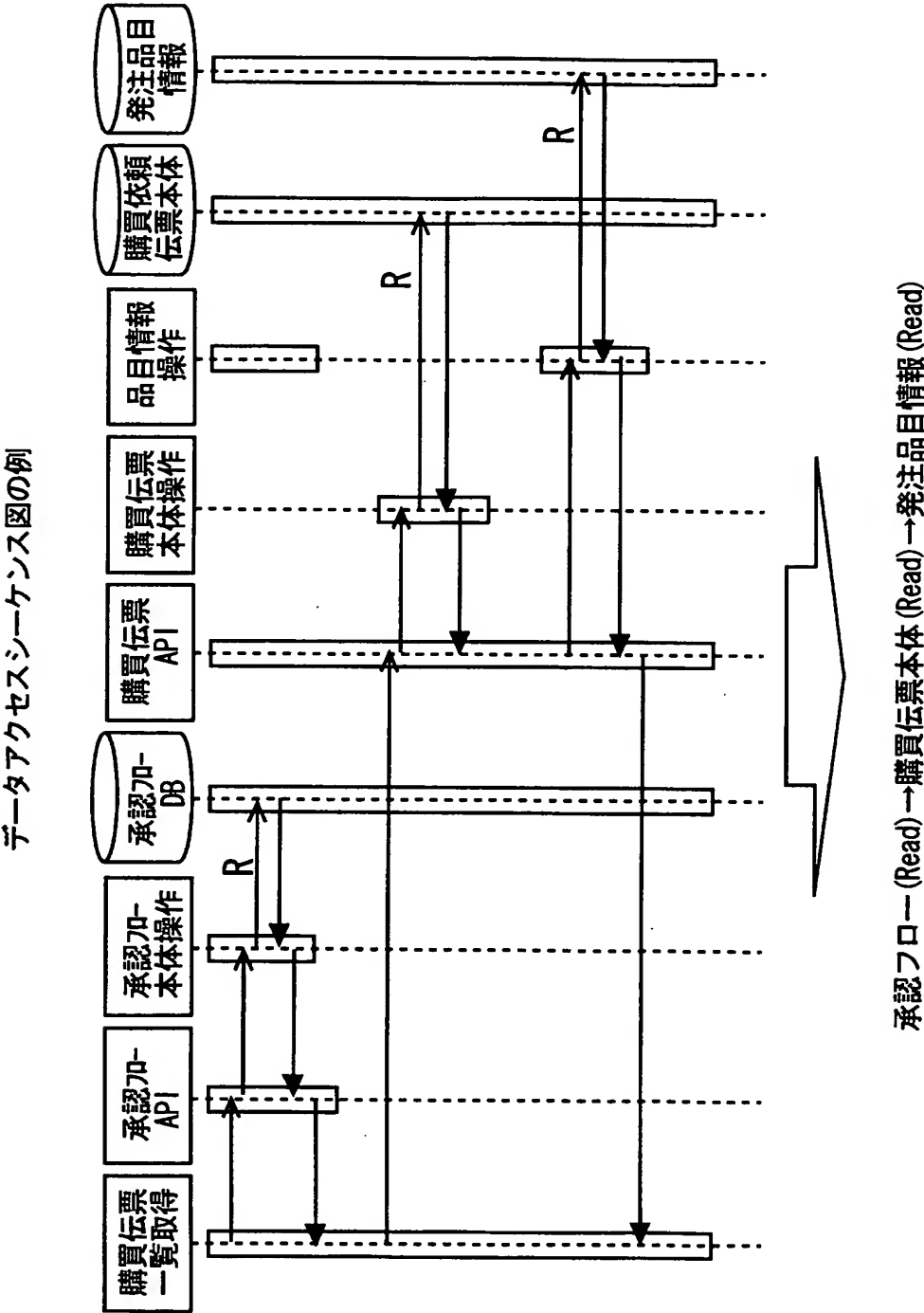
処理ルートに対するシーケンス抽出処理



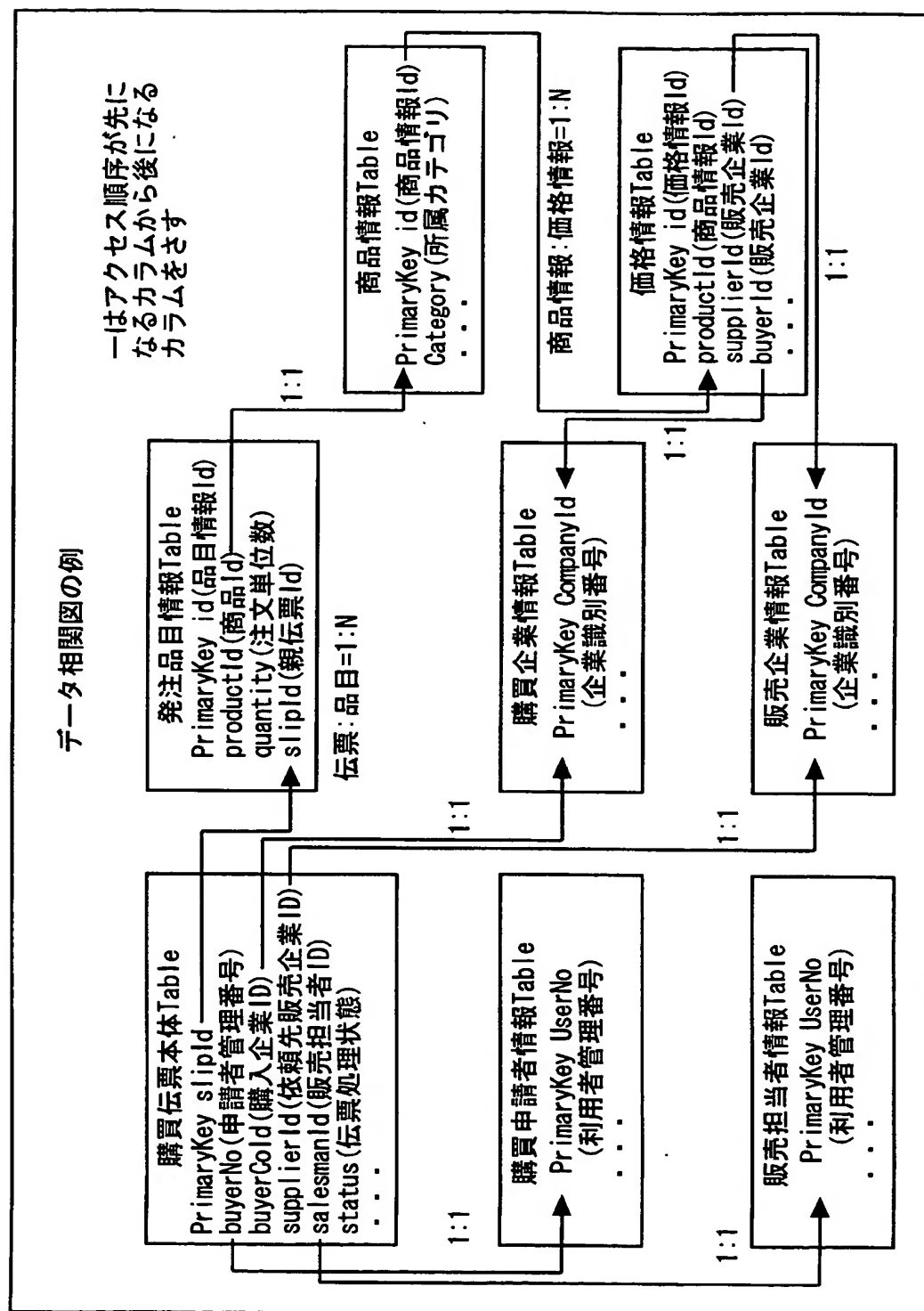
【図 5】



【図 6】



【圖 7】



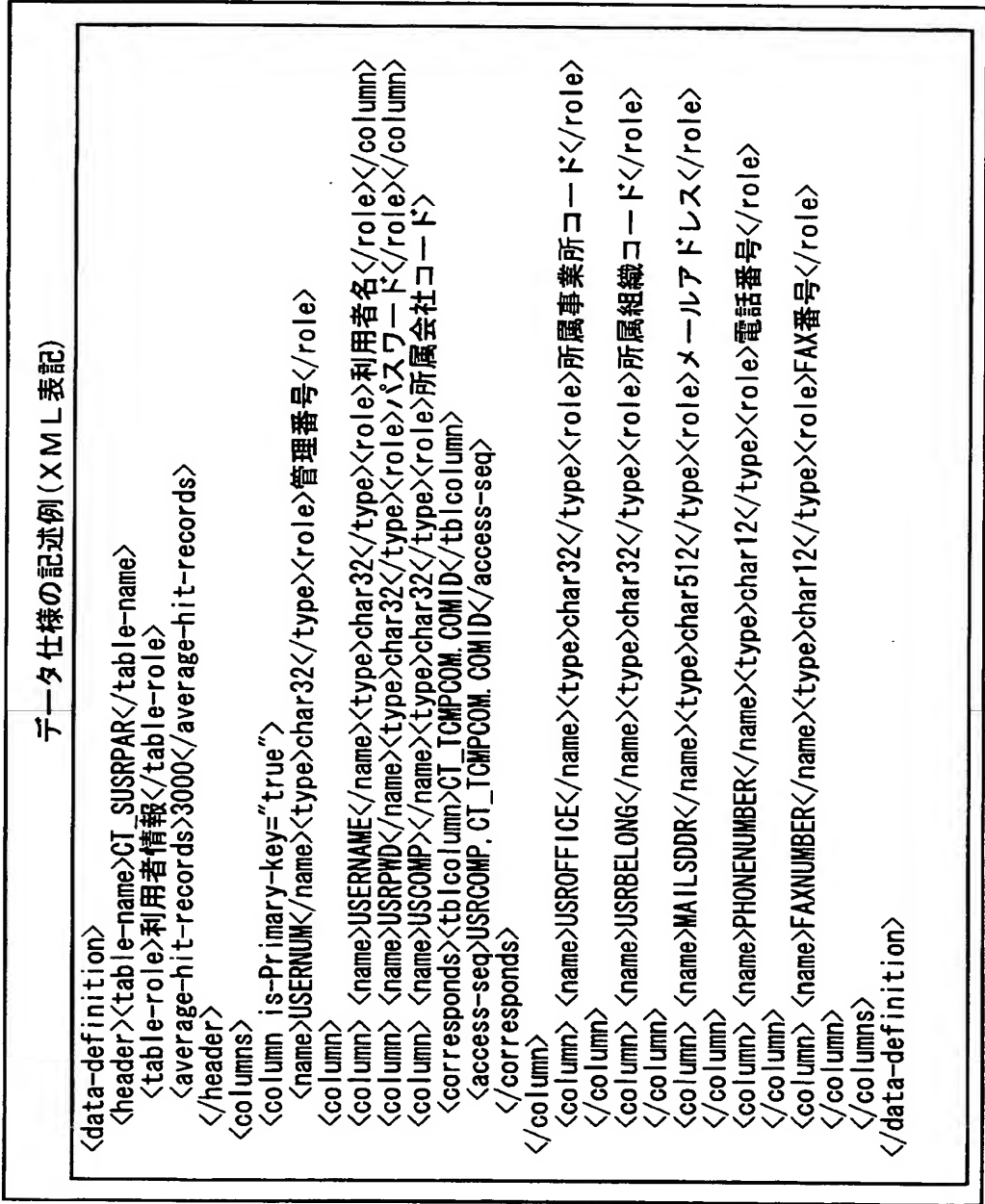
【図 8】

データ仕様の記述例			
役割:利用者情報			
テーブル名:CT_TUSPAR			
平均レコード数:3000			
カラム:USERNUM	CHAR(32)	管理番号	Primary Key
USRNAME	CHAR(32)	利用者名	
USRPWD	CHAR(32)	パスワード	
USRCOMP	CHAR(32)	所属会社コード	
CORRESPONDS CT_TCMPCOM.COMID			
TYPE=1:1			
ACCESS-SQESNCE USRCOMP.>CT_TCMPCOM.COMID			
USROFFICE	CHAR(32)	所属事業所コード	
USRBELONG	CHAR(32)	所属組織コード	
MAILADDR	CHAR(512)	メールアドレス	
PHONENUMBER	CHAR(12)	電話番号	
FAXNUMBER	CHAR(12)	FAX番号	

テーブルCT_TCMPCOMのカラムCOMIDと同じ対応は1:1で、USRCOMPを取り出し、CT_TCMPCOMにおいてCOMIDがこれと一致するレコードを参照

計算機上の実現形式としては、XMLで記述、もしくは、XML記述から生成したJava, C++などのクラスを使用する

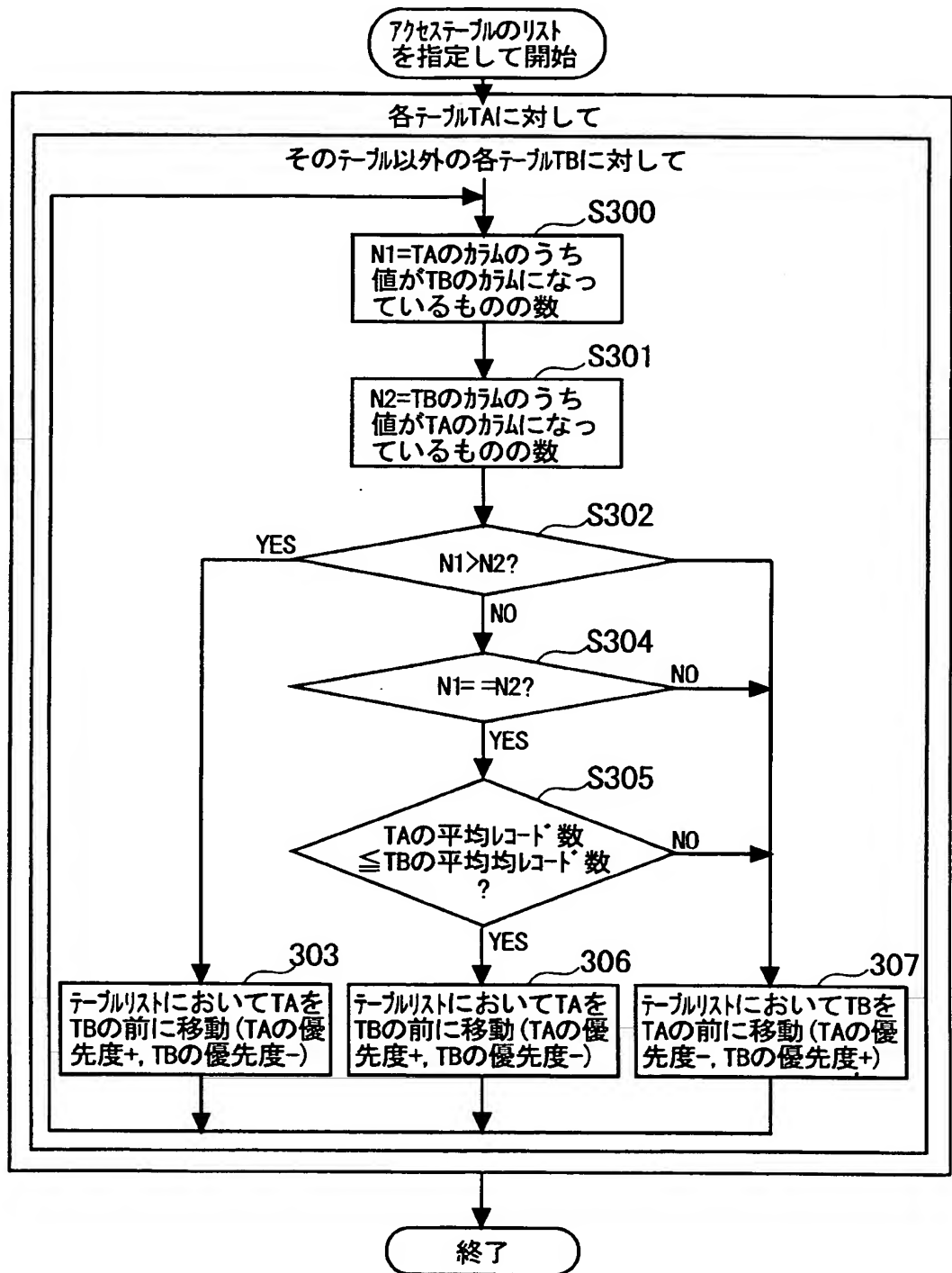
【図 9】



ヘッダ部分にテーブル名、
役割、平均ヒットコード
数を記述
カラム部分にカラム名、カラム
の型と長さ、役割、
そのカラムと同じ役割を
持つ他のカラム名
カラムのアクセス順序など
を記述する

【図 10】

標準データアクセスシーケンスを生成する処理
(標準アクセス順序抽出手順の例)



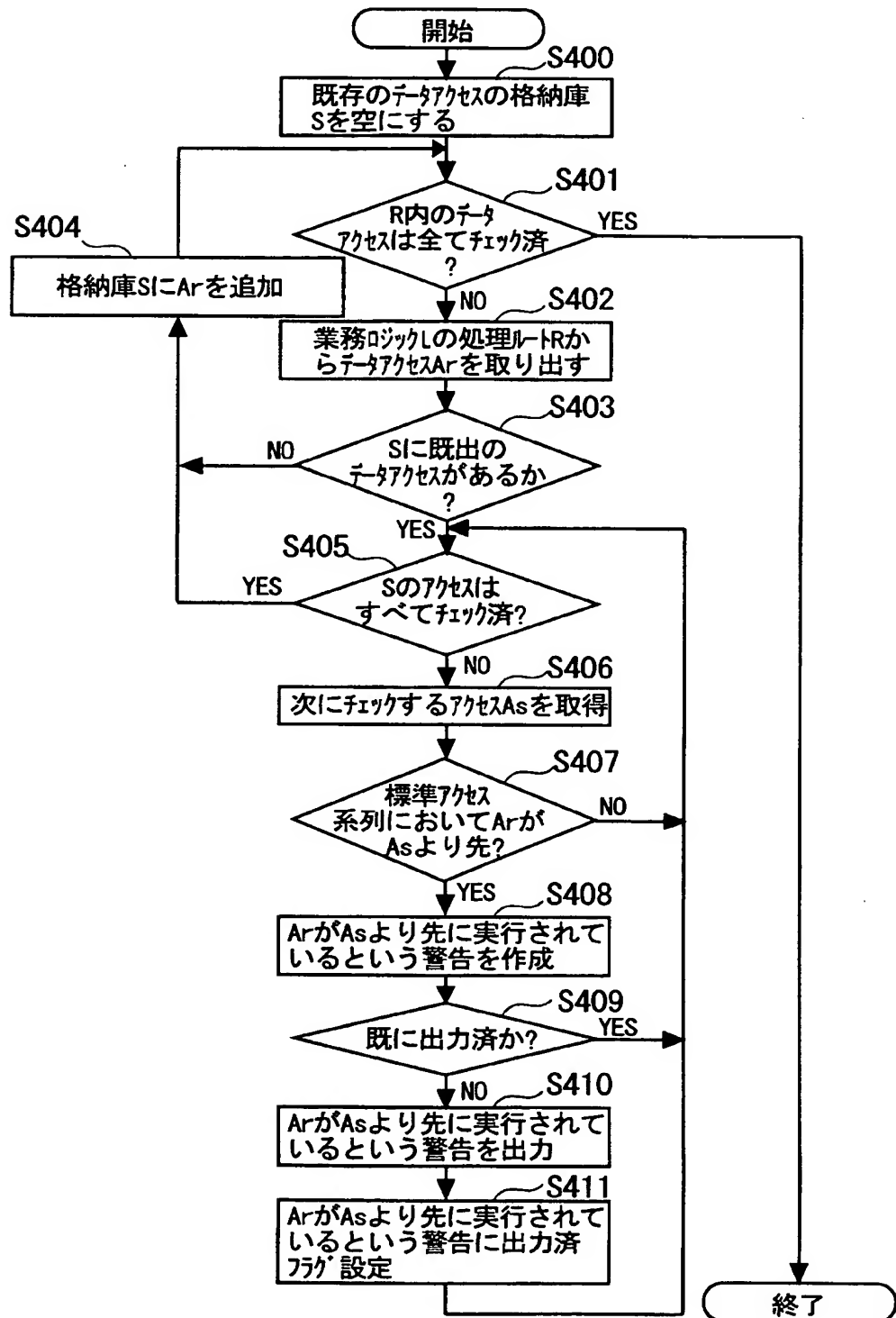
【図 11】

標準データアクセスシーケンスの例		
テーブル名	順位	
購買依頼伝票	1	
品目情報	2	
購買申請者情報	3	平均300件ヒット
販売担当者情報	3	平均1500件ヒット
商品情報	5	平均3000件ヒット
価格情報	6	
購買企業情報	7	
販売企業情報	8	

図11の手順にしたがって各テーブルの優先順位をつけると
 購買伝票情報>品目情報>商品情報>価格情報>購買企業情報、
 販売企業情報
 購買依頼伝票>購買担当者情報>購買企業情報
 購買依頼伝票>販売担当者情報>販売企業情報
 という前後関係が得られるが、購買申請者情報と販売申請者情報、
 購買申請者/販売担当者情報と
 商品情報/価格情報の前後関係がこれだけでは決まらない。そこで、
 レコードヒット件数の平均値等のヒューリスティックを使って前後
 関係をつける。ここでは購買申請者の方が販売申請者よりヒットする
 レコード件数が少ない、購買申請者、販売担当者の方が商品より
 ヒットするレコード件数の平均値が少ないという
 ヒューリスティックで順位をつけている。

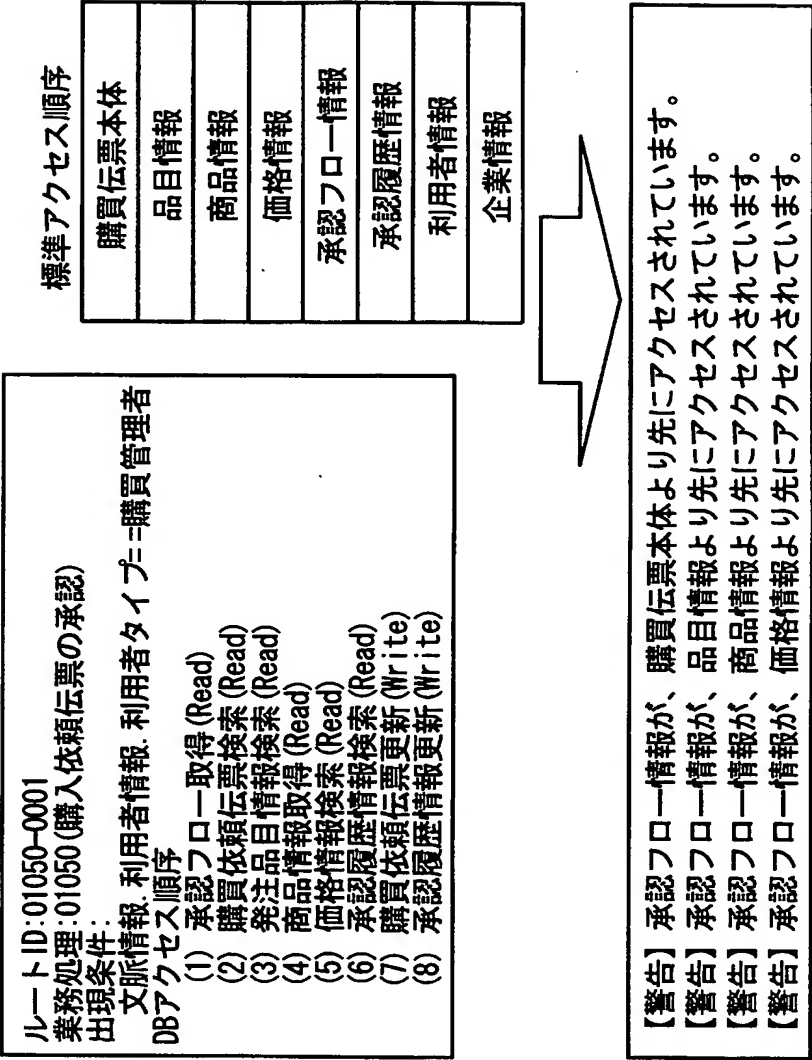
【図 12】

逸脱情報を生成する処理
(標準アクセス順序との比較)



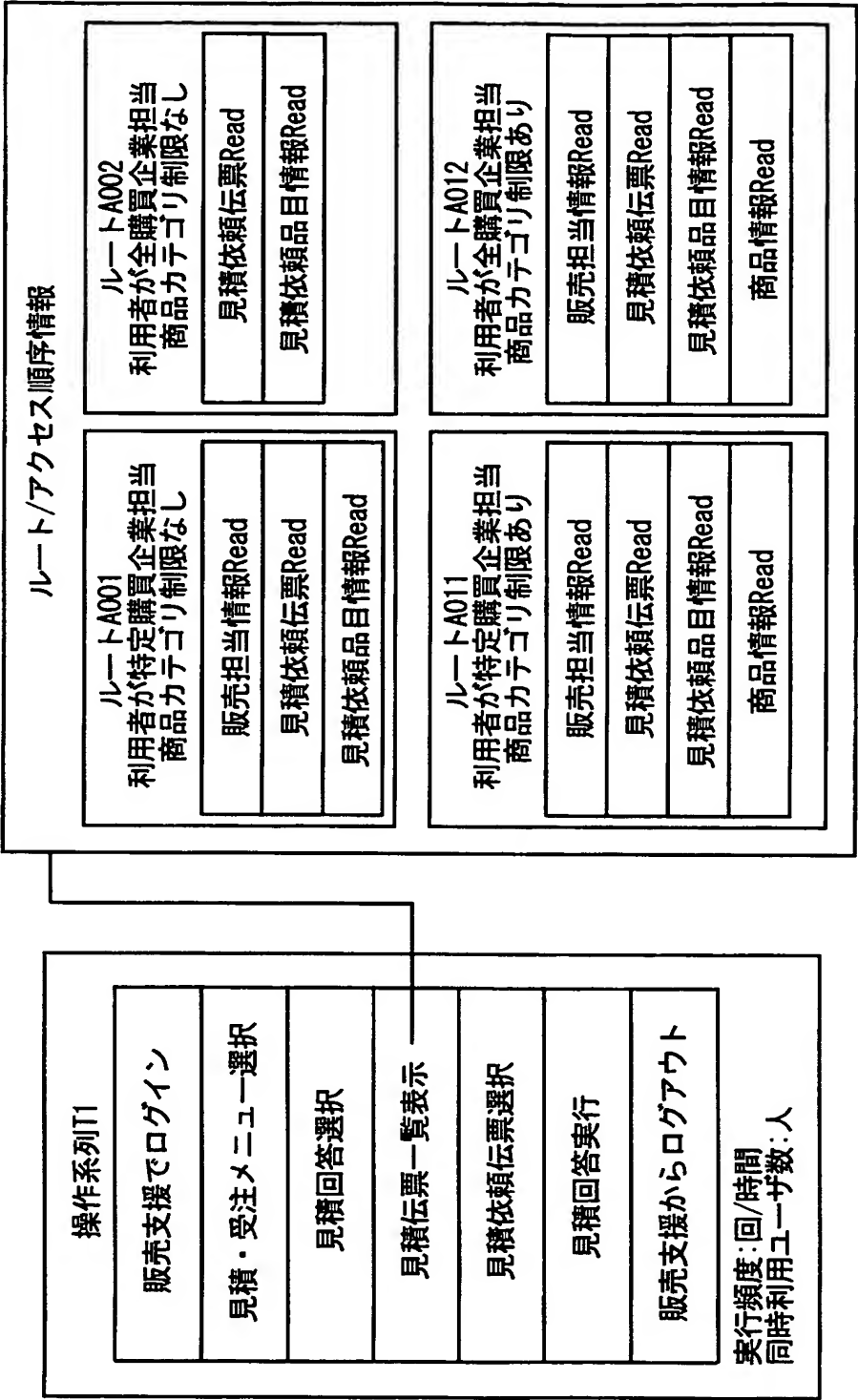
【図 13】

標準順序からの逸脱検出結果の例
(逸脱情報等の例)

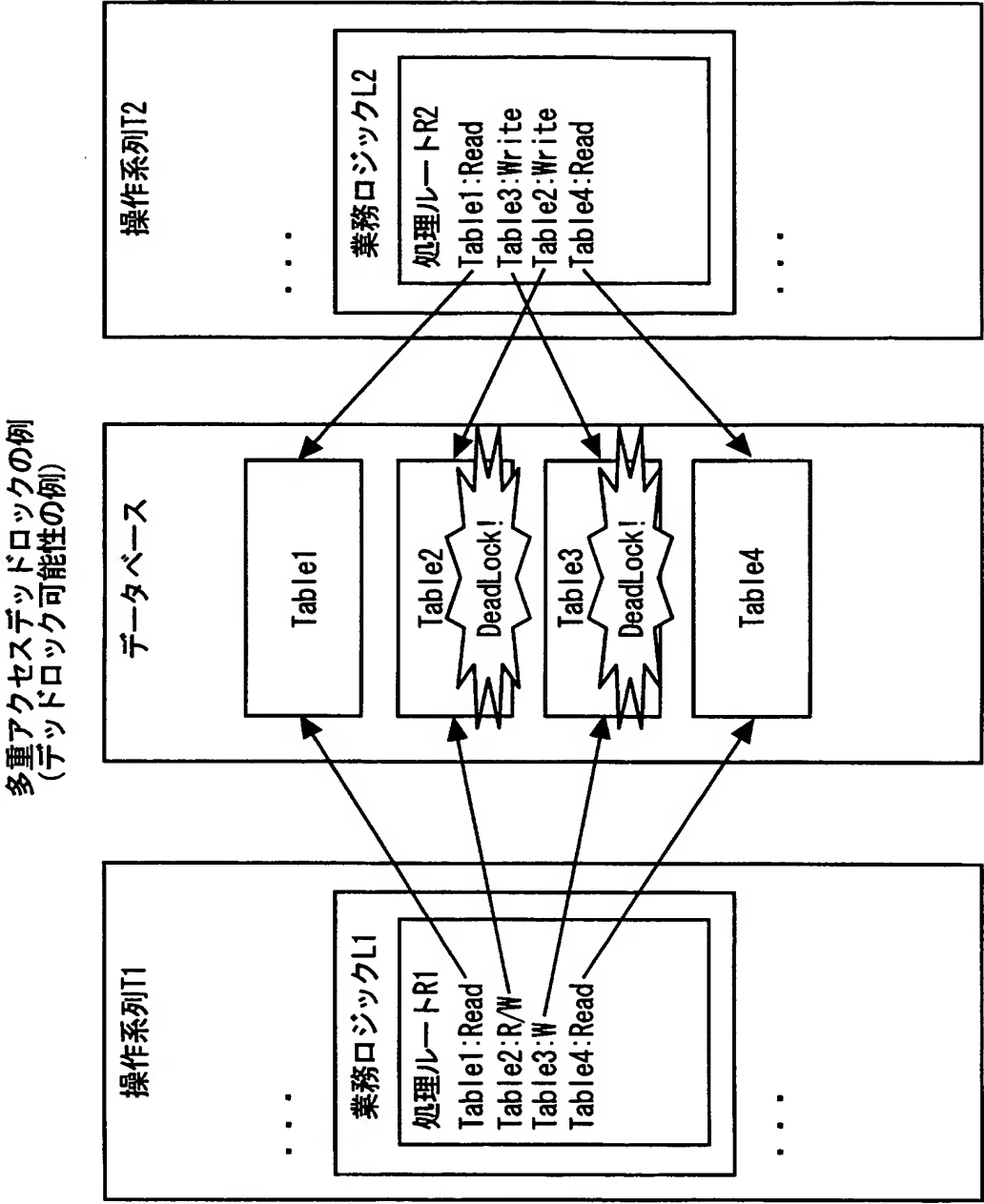


【図 14】

多重アクセスのための操作系列記述の例

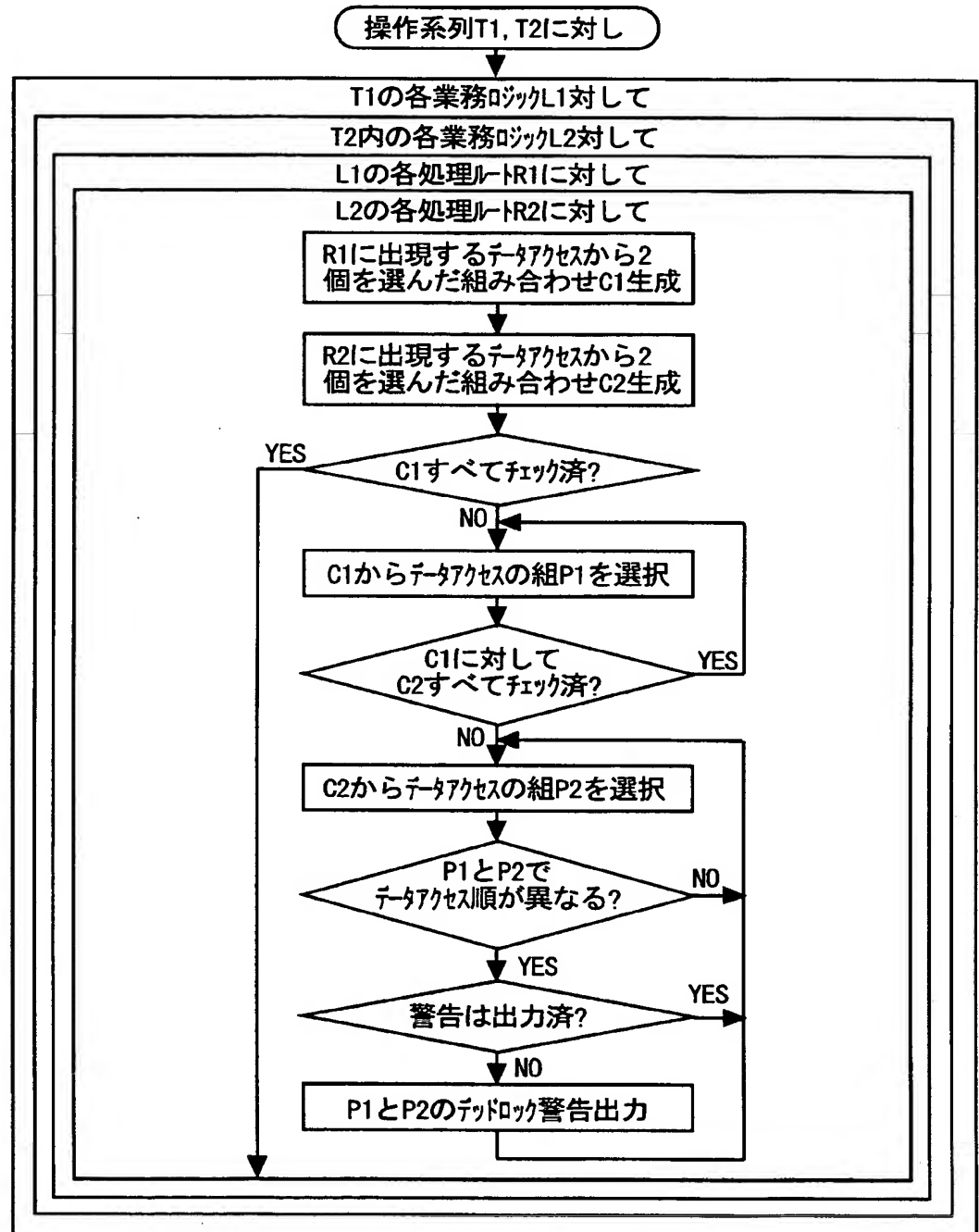


【図 15】



【図 16】

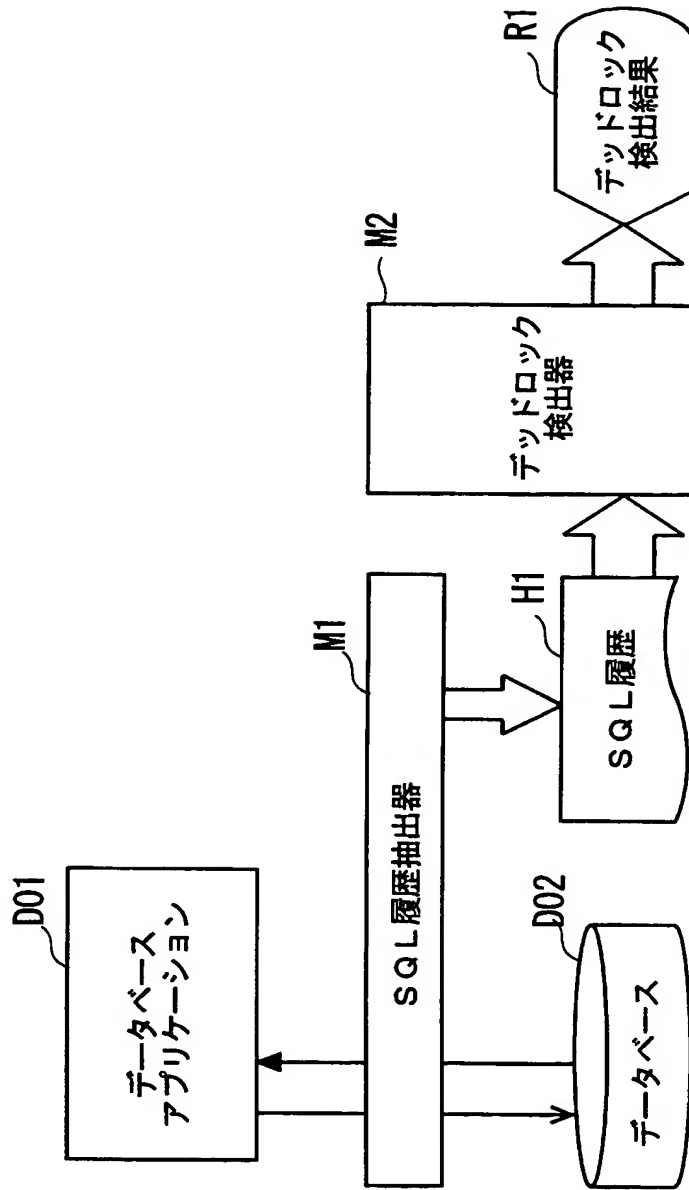
多重アクセス時のデッドロックを
検出するアルゴリズム
(多重アクセスデッドロック検出手順)



【図 17】

従来のデッドロック検出方法

(SQL発行からデッドロックの可能性抽出)



【書類名】 要約書**【要約】**

【課題】 デッドロック検出対象のプログラムを実際に動作させることなく、事前にデッドロックの可能性を検出する。

【解決手段】 コンピュータに、複数のデータベースのうちのいずれかへのアクセスを伴うアクセスステップを含む複数の処理ステップにより構成される業務ロジック設計情報を読み込ませる第1手順、前記業務ロジック設計情報に基づいて、少なくとも2つのアクセスステップにより構成される処理ルートを生成する第2手順、前記処理ルートから、第1アクセスステップ、及び第2アクセスステップを取得する第3手順、前記第1及び第2アクセスステップそれぞれによるデータベースへのアクセス順序が予め定められたアクセス順序か否かを判定する第4手順、前記アクセス順序が予め定められたアクセス順序でないと判定された場合に、予め定められたアクセス順序から逸脱した旨を報知する第5手順、を実行させるためのデッドロック事前検出プログラム。

【選択図】 図1

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 3 5 8 2 6 8
受付番号	5 0 3 0 1 7 2 9 8 8 4
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 1 0 月 2 0 日

< 認定情報・付加情報 >

【提出日】 平成15年10月17日

特願 2 0 0 3 - 3 5 8 2 6 8

出 願 人 履 歷 情 報

識別番号

[0 0 0 0 0 5 2 2 3]

1. 変更年月日

1 9 9 6 年 3 月 2 6 日

[変更理由]

住所変更

住 所

神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名

富士通株式会社